

1    **Amendments to the Claims**

2

3           The following listing of claims will replace all prior versions, and listings, of  
4    claims in the ancestor application.

5

6           1. (Original) A method for performing concurrent mark-sweep garbage  
7    collection, comprising:

8           receiving an application;

9           executing the application in at least one thread;

10          determining if available space in a heap falls below a threshold;

11          performing mark-sweep garbage collection, concurrently while executing  
12    the application, in a heap block of the heap using a first bit vector, a second bit  
13    vector, a mark bit vector pointer, and a sweep bit vector pointer in the heap  
14    block, if the available space falls below the threshold; and otherwise,

15          continuing executing the application and monitoring if the available space  
16    in the heap falls below the threshold, until the execution of the application is  
17    complete.

1

1           2. (Original) The method of claim 1, wherein the heap comprises at least  
2    one heap block.

1

1           3. (Original) The method of claim 1, further comprising initializing a  
2    concurrent mark-sweep garbage collector.

1

1           4. (Original) The method of claim 3, wherein initializing the concurrent  
2    mark-sweep garbage collector comprises setting each bit in the first bit vector  
3    and the second bit vector to 0, and pointing the mark bit vector pointer to the first  
4    bit vector and the sweep bit vector pointer to the second bit vector.

1

1           5. (Original) The method of claim 1, wherein performing mark-sweep  
2 garbage collection comprises:  
3           invoking at least one garbage collection thread to trace live objects in the  
4 heap block concurrently while executing the application; and  
5           reclaiming storage space occupied by objects other than the live objects in  
6 the block concurrently while tracing the live objects in the block and executing the  
7 application.

1  
1           6. (Original) The method of claim 5, wherein tracing the live objects in the  
2 heap block comprises parallel marking the live objects by at least one garbage  
3 collection thread.

1  
1           7. (Original) The method of claim 6, wherein parallel marking the live  
2 objects comprises setting bits corresponding to starting addresses of the live  
3 objects in a bit vector pointed to by the mark bit vector pointer to 1, by the at least  
4 one garbage collection thread.

1  
1           8. (Original) The method of claim 5, wherein reclaiming the storage space  
2 occupied by objects other than the live objects in the heap block comprises  
3 sweeping the heap block to make the said storage space allocable by using a bit  
4 vector pointed to by the sweep bit vector pointer.

1  
1           9. (Original) The method of claim 6, further comprising toggling a bit vector  
2 pointed to by the mark bit vector pointer with a bit vector pointed to by the sweep  
3 bit vector pointer after marking the live objects in the heap block is complete.

1  
1           10. (Original) The method of claim 8, further comprising setting the bit  
2 vector back to 0 after completing sweeping the heap block.

1 11. (Original) The method of claim 1, further comprising performing  
2 another cycle of concurrent mark-sweep garbage collection when available  
3 space in the heap falls below the threshold again.

1 12. (Original) A method for automatically collecting garbage objects,  
2 comprising:  
3 receiving a first code;  
4 compiling the first code into a second code;  
5 executing the second code in at least one thread; and  
6 automatically performing mark-sweep garbage collection using bit vector  
7 toggling, concurrently with the executing second code, to ensure there is storage  
8 space available for executing the second code.

1 13. (Original) The method of claim 12, wherein automatically performing  
2 mark-sweep garbage collection using bit vector toggling comprises detecting if  
3 available space in a heap falls below a threshold and invoking the concurrent  
4 mark-sweep garbage collection when the available space falls below the  
5 threshold.

1 14. (Original) The method of claim 13, wherein the heap comprises at  
2 least one heap block.

1 15. (Original) The method of claim 13, further comprising using two bit  
2 vectors for a heap block, one for marking and the other for sweeping, and  
3 toggling the two bit vectors after marking phase for the heap block is complete.

1 16. (Original) A system for concurrent mark-sweep garbage collection,  
2 comprising:  
3 a root set enumeration mechanism to enumerate references to live objects  
4 in a heap;

5 a live object tracing mechanism to parallel trace live objects in a heap  
6 block and mark the live objects in a first bit vector pointed to by a mark bit vector  
7 pointer in the heap block, concurrently with execution of an application; and  
8 a garbage sweeping mechanism to sweep storage space occupied by  
9 garbage objects to make the storage space allocable using a second bit vector  
10 pointed to by a sweep bit vector pointer in the heap block, concurrently with the  
11 execution of the application and live object marking.

1  
1 17. (Original) The system of claim 16, further comprising a bit vector  
2 toggling mechanism to toggle the first bit vector pointed to by the mark bit vector  
3 pointer and the second bit vector pointed to by the sweep bit vector pointer in the  
4 heap block.

1  
1 18. (Original) The system of claim 16, wherein the live object tracing  
2 mechanism comprises:

3 a live object search mechanism to parallel search live objects in a heap  
4 block by at least one garbage collection thread;

5 a live object marking mechanism to parallel mark the live objects in a bit  
6 vector stored in the heap block by the at least one garbage collection thread;

7 a live object scanning mechanism to parallel scan any objects reachable  
8 from the live objects in the heap; and

9 a conflict prevention mechanism to prevent more than one garbage  
10 collection thread from marking the same object.

1  
1 19. (Original) A managed runtime system, comprising:

2 a just-in-time compiler to compile an application into a code native to an  
3 underlying computing platform;

4 a virtual machine to execute the application; and

5 a garbage collector to trace live objects, mark the live objects in a first bit  
6 vector pointed to by a mark bit vector pointer in a heap block of a heap, and  
7 toggle the bit first vector pointed to by the mark bit vector pointer with a second

8 bit vector pointed to by a sweep bit vector pointer at the end of marking phase,  
9 concurrently with execution of the application.

1

1 20. (Original) The system of claim 19, further comprising a garbage  
2 sweeping mechanism to sweep storage space occupied by garbage objects to  
3 make the storage space allocable using a bit vector pointed to by the sweep bit  
4 vector pointer, concurrently with the execution of the application and live objects  
5 marking.

1

1 21. (Original) The system of claim 19, wherein the garbage collector  
2 comprises:

3 a live object marking mechanism to parallel mark the live objects in the  
4 first bit vector pointed to by the mark bit vector in the heap block of the heap; and  
5 a bit vector toggling mechanism to toggle the first bit vector pointed to by  
6 the mark bit vector pointer and the second bit vector pointed to by the sweep bit  
7 vector pointer.

1

1 22-24. (Cancelled)

1 25. (Original) An article comprising: a machine accessible medium having  
2 content stored thereon, wherein when the content is accessed by a processor,  
3 the content provides for performing concurrent mark-sweep garbage collection  
4 by:

5 receiving an application;  
6 executing the application in at least one thread;  
7 determining if available space in a heap falls below a threshold;  
8 performing mark-sweep garbage collection, concurrently while executing  
9 the application, in a heap block of the heap using a first bit vector, a second bit  
10 vector, a mark bit vector pointer, and a sweep bit vector pointer in the heap  
11 block, if the available space falls below the threshold; and otherwise,

continuing executing the application and monitoring if the available space in the heap falls below the threshold, until the execution of the application is complete.

26. (Original) The article of claim 25, wherein the heap comprises at least one heap block.

27. (Original) The article of claim 25, further comprising content for initializing a concurrent mark-sweep garbage collector.

28. (Original) The article of claim 27, wherein content for initializing the concurrent mark-sweep garbage collector comprises content for setting each bit in the first bit vector and the second bit vector to 0, and pointing the mark bit vector pointer to the first bit vector and the sweep bit vector pointer to the second bit vector.

29. (Original) The article of claim 25, wherein content for performing mark-sweep garbage collection comprises content for:  
invoking at least one garbage collection thread to trace live objects in the heap block concurrently while executing the application; and  
reclaiming storage space occupied by objects other than the live objects in the block concurrently while tracing the live objects in the block and executing the application.

30. (Original) The article of claim 29, wherein content for tracing the live objects in the heap block comprises content for parallel marking the live objects by at least one garbage collection thread.

31. (Original) The article of claim 30, wherein content parallel marking the live objects comprises content for setting bits corresponding to starting

3 addresses of the live objects in a bit vector pointed to by the mark bit vector  
4 pointer to 1, by the at least one garbage collection thread.

1

1 32. (Original) The article of claim 29, wherein content for reclaiming the  
2 storage space occupied by objects other than the live objects in the heap block  
3 comprises content sweeping the heap block to make the said storage space  
4 allocable by using a bit vector pointed to by the sweep bit vector pointer.

1

1 33. (Original) The article of claim 30, further comprising content for  
2 toggling a bit vector pointed to by the mark bit vector pointer with a bit vector  
3 pointed to by the sweep bit vector pointer after marking the live objects in the  
4 heap block is complete.

1

1 34. (Original) The article of claim 32, further comprising content for setting  
2 the bit vector back to 0 after completing sweeping the heap block.

1

1 35. (Original) The article of claim 25, further comprising content  
2 performing another cycle of concurrent mark-sweep garbage collection when  
3 available space in the heap falls below the threshold again.

1

1 36. (Original) An article comprising: a machine accessible medium having  
2 content stored thereon, wherein when the content is accessed by a processor,  
3 the content provides for automatically collecting garbage objects by:

4 receiving a first code;

5 compiling the first code into a second code;

6 executing the second code in at least one thread; and

7 automatically performing mark-sweep garbage collection using bit vector  
8 toggling, concurrently with the executing second code, to ensure there is storage  
9 space available for executing the second code.

37. (Original) The article of claim 36, wherein content for automatically performing mark-sweep garbage collection using bit vector toggling comprises content for detecting if available space in a heap falls below a threshold and invoking the concurrent mark-sweep garbage collection when the available space falls below the threshold.

38. (Original) The article of claim 37, wherein the heap comprises at least one heap block.

39. (Original) The article of claim 37, further comprising content for using two bit vectors for a heap block, one for marking and the other for sweeping, and toggling the two bit vectors after marking phase for the heap block is complete.